

Nvidia Flare User Guide for Site Admin

Version 1.0

Author: Shuiliang (Leon) Wu

Table of Contents

Nvidia Flare User Guide for Site Admin

Version 1.0

Table of Contents

Introduction

Obtain Approval

What You Have Before Started

Getting Started

Prepare Dataset

System Requirements

For Windows 11

Install WSL2

Enable WSL

Verify And Update WSL Installation

Launch Ubuntu

Install Docker

Configure GPU

Install NVIDIA Driver

Setup CUDA

Install NVIDIA Container Toolkit

Configure with Docker

Verification

Download Nvflare_mac Image From Dockerhub

Setup Workspaces

Create Site Docker Container(s)

For Site(s) Equipped with GPU

For Site(s) without GPU

Launch NVFLARE Site(s)

Access Results

For macOS

Install Docker Desktop

Download Nvflare_mac Image From Dockerhub

Setup Workspaces

Create Site Docker Container(s)

Launch NVFLARE Site(s)

Access Results

Troubleshooting

Common Issues

Support Resources

Appendix

Requirements of Data Format

Introduction

Welcome to the User Guide for [Federated Learning Framework Nvidia Flare \(nvflare\)](#). This guide is written for **Site Admin(s)** participating Federated Learning projects. It will help you as the Site Admin to setup nvflare in your computer and connect to Server for federated learning on your data.

Obtain Approval

To get approval on your site(s), you as site admin is required to **send an request together with site(s) information to project manager**. Once approved by project manager, your site admin account will be setup based on your shared site(s) information including items below:

- Organization name
- Site name(s) and their(its) capacity (numbers of GPU and memory per GPU)

Demo examples of site(s) information to be sent:

From site admin University of Calgary:

Organization Name	Site Name	Numbers of GPU	Memory per GPU
University of Calgary	site1_ucalgary_win11	1	24
University of Calgary	site2_ucalgary_win11	0	0

From site admin St. Joseph's Healthcare:

Organization Name	Site Name	Numbers of GPU	Memory per GPU
St. Joseph's Healthcare	site_stjoes_mac	0	0

What You Have Before Started

- Approval from project manager
- Nvidia Flare User Guide for Site Admin (this one)
- Raw datasets (from your own site(s))
- Package from project manager including:
 1. ip address of Server. Here it is `192.168.2.100` as the demo.
 2. Access to download nvflare_mac docker image. Here it is `docker pull shuiliangwu/nvflare_mac:latest` or click [Docker Hub Repository](#).
 3. Site startup kits.
 4. `jobs` folder.

Getting Started

Prepare Dataset

Prepare dataset as per [Requirements of Data Format](#).

Formatted datasets for this demo can be downloaded from [here](#) in the folders of `package_for_site_admin_stjoes` and `package_for_site_admin_ucalgary`.

System Requirements

- Operating System: Windows 11 / macOS
- Memory: 4GB RAM minimum
- Disk Space: 30GB free space minimum
- Internet Connection

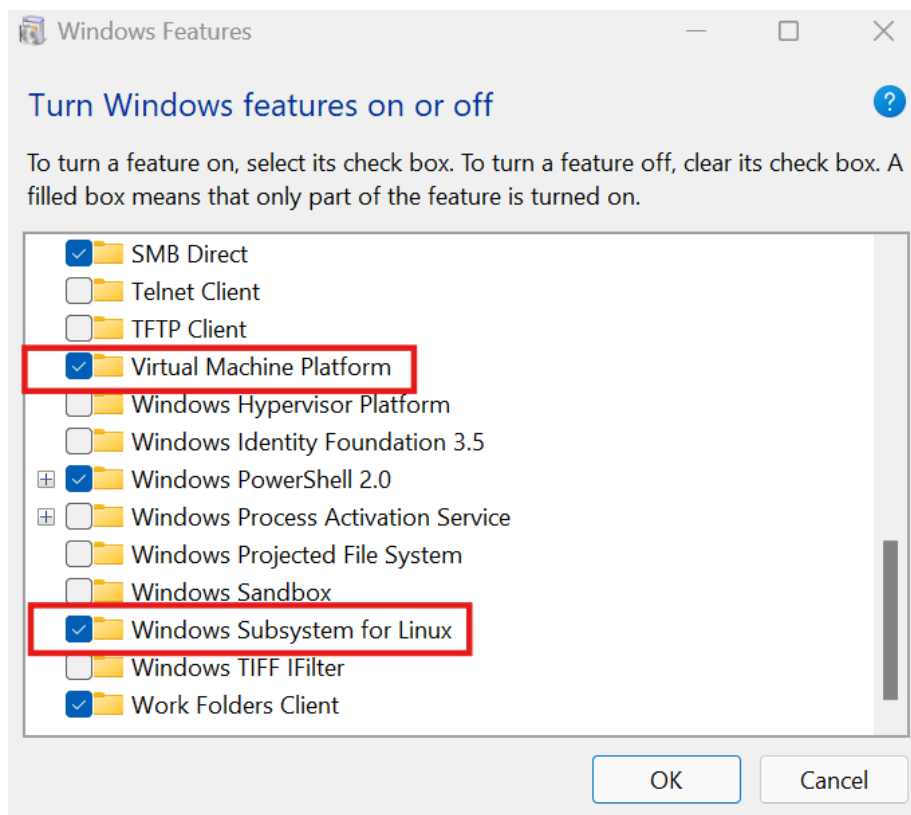
This User Guide is demonstrated using both [Windows 11](#) and [macOS](#) (click to jump to the right section to start).

For Windows 11

Install WSL2

Enable WSL

1. Press `win + R` to open the Run dialog box.
2. Type `optionalfeatures` and press Enter to open the Windows Features dialog.
3. In the Windows Features dialog, scroll down and check the box next to `Virtual Machine Platform` and `Windows Subsystem for Linux`.



4. Click **OK** to apply the changes and restart your computer when prompted.
5. Press **win + X** and select Windows Terminal (Admin) or PowerShell (Admin).
6. Run the WSL Install Command:
 - In the PowerShell window, type the following command and press **Enter**:

```
ws1 --install -d Ubuntu-20.04
```

```
Administrator: Windows Pow
PS C:\Users\shuil> wsl --install -d Ubuntu-20.04
Installing Windows optional component: VirtualMachinePlatform

Deployment Image Servicing and Management tool
Version: 10.0.22621.2792

Image Version: 10.0.22631.3880

Enabling feature(s)
[=====100.0%=====]
The operation completed successfully.
Installing: Ubuntu 20.04 LTS
Ubuntu 20.04 LTS has been installed.
The requested operation is successful. Changes will not be effective until the system is rebooted.
```

- Restart your PC to complete the WSL installation.

Verify And Update WSL Installation

1. Check WSL Version by running below. This will display a list of installed Linux distributions and their WSL versions. Ensure that your distribution is set to version 2.

```
ws1 --list --verbose
```

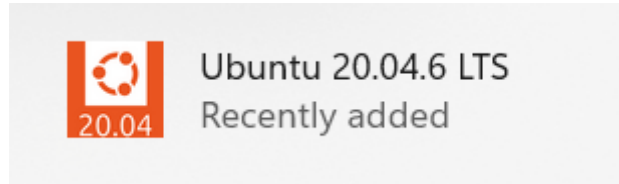
2. Ensure you have the latest WSL update by running:

```
ws1 --update
```

```
PS C:\Users\shuil> wsl --list --verbose
  NAME      STATE      VERSION
* Ubuntu    Stopped    2
PS C:\Users\shuil> wsl --update
Checking for updates.
The most recent version of Windows Subsystem for Linux is already installed.
```

Launch Ubuntu

1. Open the Start menu, find your installed distribution (i.e., Ubuntu 20.04), and click on it.

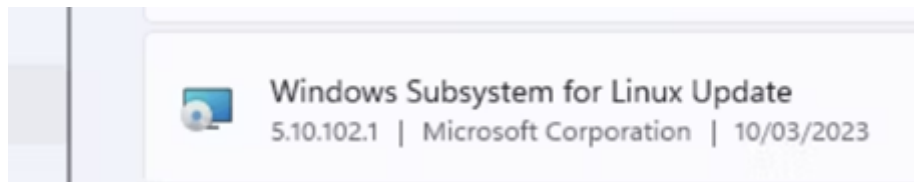


2. Follow the on-screen instructions to set up your Linux distribution, which usually involves creating a username and password.

```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: leonwu
New password:
Retype new password:
passwd: password updated successfully
The operation completed successfully.
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.153.1-microsoft-standard-WSL2 x86_64)
```

3. When you see an error of **"WSL 2 requires an update to its kernel component"** when running Ubuntu 20.04, check if below is in place in your installed apps. If not, make sure [WSL2 Linux kernel update package for x64 machines](#) is downloaded and installed.



Install Docker

1. Open Linux terminal (i.e, Ubuntu 20.04, terminal for short) and run:

```
sudo apt-get update
sudo apt-get install docker.io
```

2. Verify docker installation by running:

```
sudo docker --version
sudo docker run hello-world
```

```
leonwu@DESKTOP-AI:~$ sudo docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu2~20.04.1
leonwu@DESKTOP-AI:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:94323f3e5e09a8b9515d74337010375a456c909543e1ff1538f5116d38ab3989
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Configure GPU

Click [here](#) to skip this section if your computer is not equipped with a GPU.

Since the demo computer is equipped with GeForce RTX 3090, this section is demonstrated with GeForce RTX 3090.

Install NVIDIA Driver

1. Search right NVIDIA Driver from [here](#).

NVIDIA Driver Downloads

Select from the dropdown list below to identify the appropriate driver for your NVIDIA product.

[Help](#)

Product Type:	GeForce	▼
Product Series:	GeForce RTX 30 Series	▼
Product:	GeForce RTX 3090	▼
Operating System:	Windows 11	▼
Download Type:	Studio Driver (SD)	▼ ?
Language:	English (US)	▼

Search

2. Download NVIDIA Driver.

NVIDIA Studio Driver

Version:	555.99 ^{WHQL}
Release Date:	2024.6.4
Operating System:	Windows 10 64-bit, Windows 11
Language:	English (US)
File Size:	628.79 MB

Download

3. Install NVIDIA Driver by double-clicking the downloaded `.exe` file and following the installation instructions. **This is the only driver you need to install. Do not install any Linux display driver in WSL.**



Setup CUDA

1. Open terminal, first remove the old GPG key by running:

```
sudo apt-key del 7fa2af80
```

```
leonwu@DESKTOP-AI:~$ sudo apt-key del 7fa2af80
[sudo] password for leonwu:
OK
```

2. Install Linux x86 CUDA Toolkit using WSL-Ubuntu Package following the [Installation Instructions](#).

- Select WSL-Ubuntu Package.

Operating System	<div>Linux</div>	<div>Windows</div>					
Architecture	<div>x86_64</div>	<div>arm64-sbsa</div>	<div>aarch64-jetson</div>				
Distribution	<div>Amazon-Linux</div>	<div>Debian</div>	<div>Fedora</div>	<div>KylinOS</div>	<div>OpenSUSE</div>	<div>RHEL</div>	<div>Rocky</div>
	<div>SLES</div>	<div>Ubuntu</div>	<div>WSL-Ubuntu</div>				
Version	<div>2.0</div>						
Installer Type	<div>deb (local)</div>	<div>deb (network)</div>	<div>runfile (local)</div>				

- Run Installation Instructions below one by one:

```
wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-wsl-ubuntu.pin
sudo mv cuda-wsl-ubuntu.pin /etc/apt/preferences.d/cuda-repository-pin-600
wget https://developer.download.nvidia.com/compute/cuda/12.5.1/local_installers/cuda-repo-wsl-ubuntu-12-5-local_12.5.1-1_amd64.deb
sudo dpkg -i cuda-repo-wsl-ubuntu-12-5-local_12.5.1-1_amd64.deb
sudo cp /var/cuda-repo-wsl-ubuntu-12-5-local/cuda-*-keyring.gpg /usr/share/keyrings/
sudo apt-get update
sudo apt-get -y install cuda-toolkit-12-5
```

Install NVIDIA Container Toolkit

1. Configure the production repository by running below in terminal:

```
curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg && curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-container-toolkit.list | sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg] https://#g' | sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
```

2. Update the packages list from the repository:

```
sudo apt-get update
```

3. Install the NVIDIA Container Toolkit packages:

```
sudo apt-get install -y nvidia-container-toolkit
```

Configure with Docker

1. Configure the container runtime by running below in terminal:

```
sudo nvidia-ctk runtime configure --runtime=docker
```

```
Leonwu@DESKTOP-AI:~$ sudo nvidia-ctk runtime configure --runtime=docker
INFO[0000] Config file does not exist; using empty config
INFO[0000] Wrote updated config to /etc/docker/daemon.json
INFO[0000] It is recommended that docker daemon be restarted.
```

2. Restart the Docker daemon by running:

```
sudo systemctl restart docker
```

3. In case there are errors running `sudo systemctl restart docker` such as:


```

Leonwu@DESKTOP-AI:~$ sudo nvidia-ctk runtime configure --runtime=docker
INFO[0000] Config file does not exist; using empty config
INFO[0000] Wrote updated config to /etc/docker/daemon.json
INFO[0000] It is recommended that docker daemon be restarted.
Leonwu@DESKTOP-AI:~$ sudo systemctl restart docker
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down

```

or

Error: Cannot connect to the Docker daemon at unix:/var/run/docker.sock. Is the docker daemon running?

Solution is to run below. It should end up at the state `API listen on /var/run/docker.sock`. Now you can move to next verification step.

```
sudo dockerd
```

Verification

Verify GPU configuration by running below in terminal:

```
sudo docker run --rm --runtime=nvidia --gpus all ubuntu nvidia-smi
```

```

Leonwu@DESKTOP-AI:~$ sudo docker run --rm --runtime=nvidia --gpus all ubuntu nvidia-smi
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
9c704ecd0c69: Pull complete
Digest: sha256:2e863c44b718727c860746568e1d54afd13b2fa71b160f5cd9058fc436217b30
Status: Downloaded newer image for ubuntu:latest
Wed Jul 10 06:48:03 2024
+-----+
| NVIDIA-SMI 555.52.01                  Driver Version: 555.99          CUDA Version: 12.5          |
+-----+-----+-----+-----+-----+-----+
| GPU   Name                               Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC | |
| Fan  Temp  Perf    Pwr:Usage/Cap       |              |         Memory-Usage | GPU-Util  Compute M. |
|                               |              |            MIG M.     |                      |
+-----+-----+-----+-----+-----+-----+
| 0     NVIDIA GeForce RTX 3090           On           | 00000000:01:00.0 On  |           2%        N/A | |
| 0%    59C    P8             40W / 370W | 1272MiB / 24576MiB |                      Default |
|                               |              |                        N/A     |                      |
+-----+-----+-----+-----+-----+-----+
+-----+
| Processes: |
| GPU   GI   CI        PID   Type   Process name                      GPU Memory |
|   ID   ID   ID             |              |           Usage     |
+-----+-----+-----+-----+-----+
| 0     N/A  N/A         26      G      /Xwayland                          N/A        |
+-----+

```

Download Nvflare_mac Image From Dockerhub

1. In terminal, download [nvflare_mac](#) docker image by running:

```

sudo su
docker pull shuiliangwu/nvflare_mac:latest

```

2. Verify the download of nvflare_mac image by running:

```
docker images
```

```
leonwu@DESKTOP-AI:~$ sudo su
root@DESKTOP-AI:/home/leonwu# docker pull shuiliangwu/nvflare_mac:latest
latest: Pulling from shuiliangwu/nvflare_mac
9ea8908f4765: Pull complete
75ffc57cea9c: Pull complete
5c8fc1c86705: Pull complete
2f83981de011: Pull complete
Digest: sha256:52f891f94ffee46ea1c0c2ce9e746efa98b512ccfcf3c7951dcb0ed9fa395211
Status: Downloaded newer image for shuiliangwu/nvflare_mac:latest
docker.io/shuiliangwu/nvflare_mac:latest
root@DESKTOP-AI:/home/leonwu# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
shuiliangwu/nvflare_mac	latest	4e904631e435	4 days ago	9.25GB
ubuntu	latest	35a88802559d	4 weeks ago	78.1MB
hello-world	latest	d2c94e258dcb	14 months ago	13.3kB

Setup Workspaces

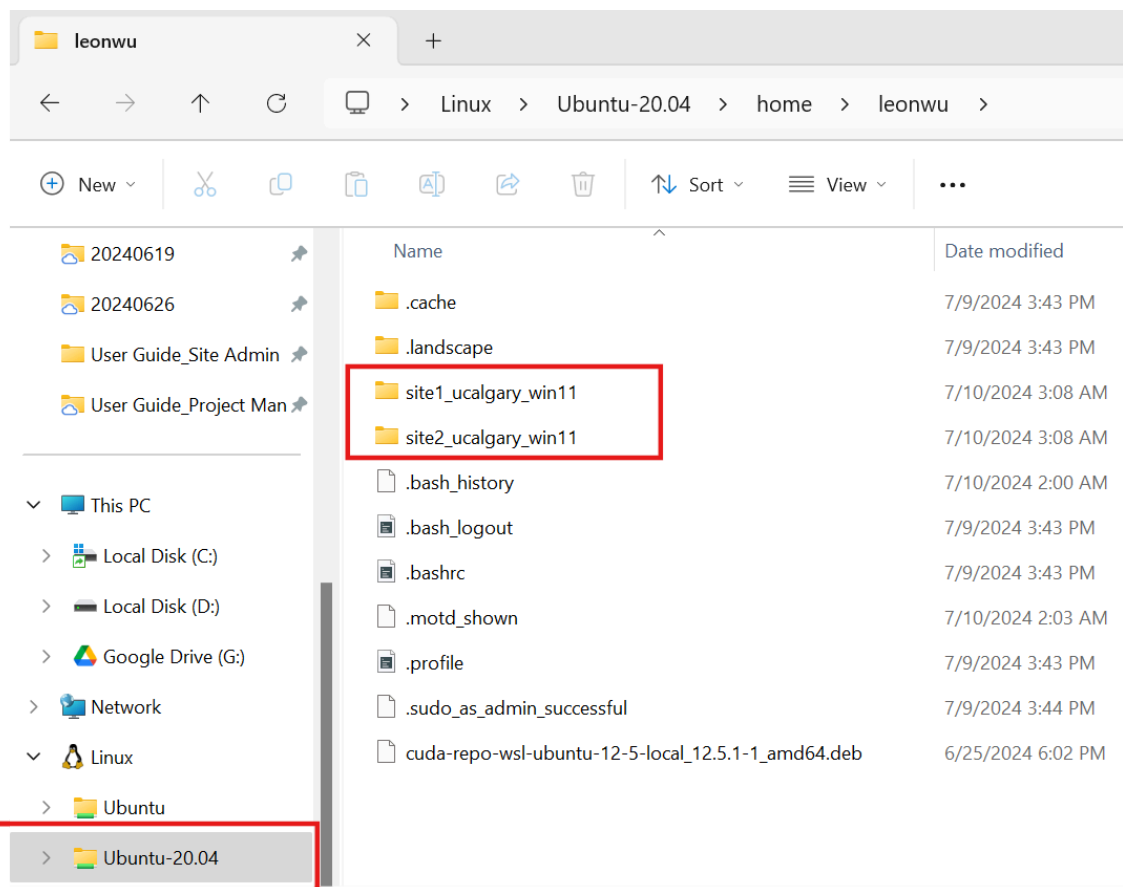
1. For the sake of convenience, let's create Site docker containers in user `leonwu` folder by running (For any scripts in this Guide that contain the placeholder `leonwu`, make sure to replace `leonwu` with your own Ubuntu account name before using them):

```
mkdir /home/leonwu/site1_ucalgary_win11
mkdir /home/leonwu/site1_ucalgary_win11/my-workspace
chmod -R 777 /home/leonwu/site1_ucalgary_win11/my-workspace

mkdir /home/leonwu/site2_ucalgary_win11
mkdir /home/leonwu/site2_ucalgary_win11/my-workspace
chmod -R 777 /home/leonwu/site2_ucalgary_win11/my-workspace
```

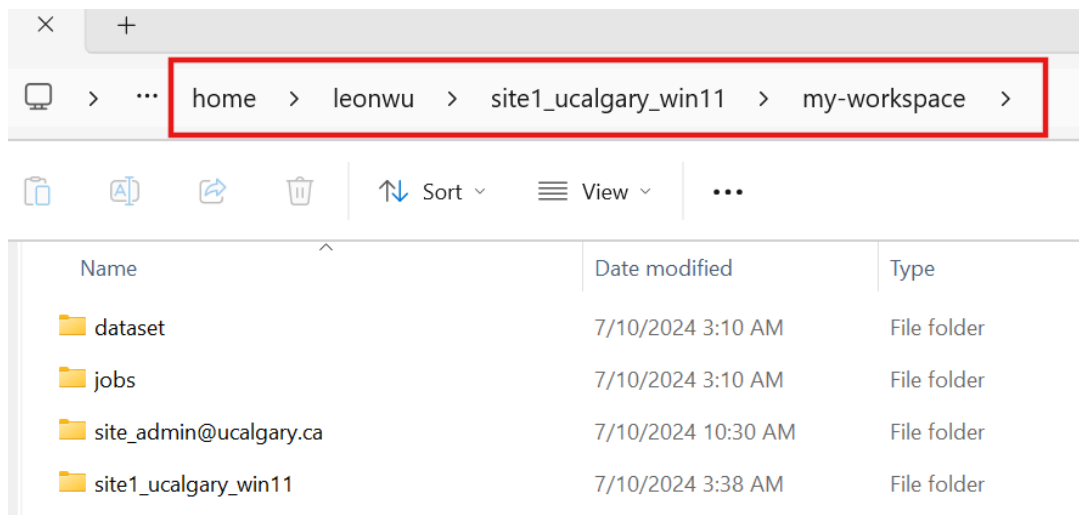
```
root@DESKTOP-AI:/home/leonwu# mkdir /home/leonwu/site1_ucalgary_win11
root@DESKTOP-AI:/home/leonwu# mkdir /home/leonwu/site1_ucalgary_win11/my-workspace
root@DESKTOP-AI:/home/leonwu# chmod -R 777 /home/leonwu/site1_ucalgary_win11/my-workspace
root@DESKTOP-AI:/home/leonwu# mkdir /home/leonwu/site2_ucalgary_win11
root@DESKTOP-AI:/home/leonwu# mkdir /home/leonwu/site2_ucalgary_win11/my-workspace
root@DESKTOP-AI:/home/leonwu# chmod -R 777 /home/leonwu/site2_ucalgary_win11/my-workspace
```

We shall see `site1_ucalgary_win11` and `site2_ucalgary_win11` folders in `Linux/Ubuntu-20.04/home/leonwu` with `my-workspace` subfolder in each of them.



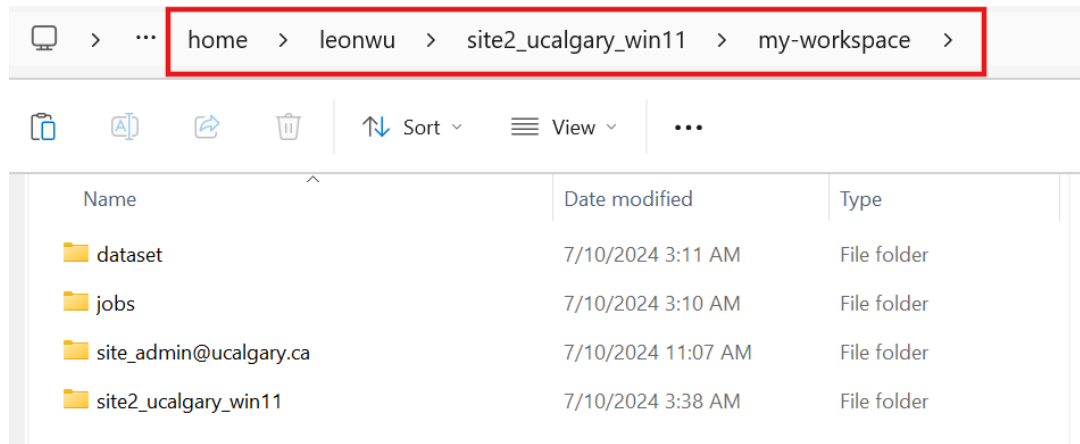
2. Copy required folders into Site(s)'s `my-workspace` folder.

- Copy below four folders into new created `my-workspace` folder of `site1_ucalgary_win11`.
 - `site_admin@ucalgary.ca` folder: site FLARE site console kit (from project manager)
 - `site1_ucalgary_win11` folder: site startup kit (from project manager)
 - `jobs` folder: contains source code and configuration files (from project manager)
 - `dataset` folder: `site1_ucalgary_win11`'s dataset which contains `site1_ucalgary_win11.csv` (all encoded data) and `site1_ucalgary_win11_header.csv` (all features).



- Copy below four folders into new created `my-workspace` folder of `site2_ucalgary_win11`.

- `site_admin@ucalgary.ca` folder: site FLARE site console kit (from project manager)
- `site2_ucalgary_win11` folder: site startup kit (from project manager)
- `jobs` folder: contains source code and configuration files (from project manager)
- `dataset` folder: `site2_ucalgary_win11`'s dataset which contains `site2_ucalgary_win11.csv` (all encoded data) and `site2_ucalgary_win11_header.csv` (all features).



Create Site Docker Container(s)

For Site(s) Equipped with GPU

1. Go to `site1_ucalgary_win11` folder by running:

```
cd /home/leonwu/site1_ucalgary_win11
```

We shall see our current location is `site1_ucalgary_win11`.

It is crucial that we are at above location in order to mount workspace.

```
root@DESKTOP-AI:/home/leonwu# cd /home/leonwu/site1_ucalgary_win11
root@DESKTOP-AI:/home/leonwu/site1_ucalgary_win11#
```

2. Create Site docker container (**this site is equipped with GPU**) by running:

```
docker run --privileged --rm -d -it --ipc=host \
--gpus all \
--ulimit memlock=-1 --ulimit stack=67108864 \
-v "$(pwd -P)/my-workspace:/workspace/my-workspace" \
-v /var/run/docker.sock:/var/run/docker.sock \
-w /workspace/my-workspace \
--name site1_ucalgary_win11 \
shuiliangwu/nvflare_mac:latest
```

Above site name `site1_ucalgary_win11` shall be replaced with the actual ones from your setup.

```

root@DESKTOP-AI:/home/leonwu/site1_ucalgary_win11# docker run --privileged --rm -d -it --ipc=host \
ulimit memlock=-1 --gpus all \
> --ulimit memlock=-1 --ulimit stack=67108864 \
> -v "$(pwd -P)/my-workspace:/workspace/my-workspace" \
> -v /var/run/docker.sock:/var/run/docker.sock \
> -w /workspace/my-workspace \
> --name site1_ucalgary_win11 \
> shuiliangwu/nvflare_mac:latest
77f26b7624e9a514eba8fdc7e31e063942fb1e31550f3ac607919d9203da3440

```

3. Verify docker containers by running:

```
docker ps -a
```

```

root@DESKTOP-AI:/home/leonwu/site1_ucalgary_win11# docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
77f26b7624e9   shuiliangwu/nvflare_mac:latest      "/bin/bash"             2 minutes ago Up 2 minutes             site1_ucalgary_win11

```

For Site(s) without GPU

1. Go to `site1_ucalgary_win11` folder by running:

```
cd /home/leonwu/site2_ucalgary_win11
```

We shall see our current location is `site1_ucalgary_win11`.

It is crucial that we are at above location in order to mount workspace.

```

root@DESKTOP-AI:/home/leonwu# cd /home/leonwu/site2_ucalgary_win11
root@DESKTOP-AI:/home/leonwu/site2_ucalgary_win11#

```

2. Create Site docker container by running:

```

docker run --privileged --rm -d -it --ipc=host \
--ulimit memlock=-1 --ulimit stack=67108864 \
-v "$(pwd -P)/my-workspace:/workspace/my-workspace" \
-v /var/run/docker.sock:/var/run/docker.sock \
-w /workspace/my-workspace \
--name site2_ucalgary_win11 \
shuiliangwu/nvflare_mac:latest

```

Above site name `site2_ucalgary_win11` shall be replaced with the actual ones from your setup.

```

root@DESKTOP-AI:/home/leonwu/site2_ucalgary_win11# docker run --privileged --rm -d -it --ipc=host \
> --ulimit memlock=-1 --ulimit stack=67108864 \
> -v "$(pwd -P)/my-workspace:/workspace/my-workspace" \
> -v /var/run/docker.sock:/var/run/docker.sock \
> -w /workspace/my-workspace \
> --name site2_ucalgary_win11 \
> shuiliangwu/nvflare_mac:latest
5f3162866e702707d7e78ba325b5b8fa7d2743daee0dd17c2bfff76878188e7e9

```

3. Verify docker containers by running:

```
docker ps -a
```

```

root@DESKTOP-AI:/home/leonwu/site2_ucalgary_win11# docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
5f3162866e70   shuiliangwu/nvflare_mac:latest      "/bin/bash"             20 seconds ago Up 19 seconds             site2_ucalgary_win11
1269967e2df0   shuiliangwu/nvflare_mac:latest      "/bin/bash"             32 minutes ago Up 32 minutes             site1_ucalgary_win11

```

Launch NVFLARE Site(s)

1. Access to Site docker container by running:

```
docker exec -it site1_ucalgary_win11 /bin/bash
```

2. We shall see previous copied four folders by running:

```
ls
```

```
root@DESKTOP-AI:/home/leonwu/site1_ucalgary_win11# docker exec -it site1_ucalgary_win11 /bin/bash
root@77f26b7624e9:/workspace/my-workspace# ls
data  logs  site1_ucalgary_win11  site_admin@ucalgary.ca
```

3. Modify the `/etc/hosts` file of Site docker container.

- Open `/etc/hosts` file by running:

```
nano /etc/hosts
```

- Replace the hostname of `172.17.0.2` by `site1_ucalgary_win11`, and add `server`'s ip address and hostname.

```
GNU nano 4.8 /etc/hosts
127.0.0.1    localhost
::1         localhost ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
172.17.0.2   site1_ucalgary_win11
192.168.2.100 server
```

- Press `Ctrl + O`, then `Enter` to save the change. Press `Ctrl + X` to exit.
- Verify the modification by running:

```
cat /etc/hosts
```

```
root@00bf59daa070:/workspace/my-workspace# cat /etc/hosts
127.0.0.1    localhost
::1         localhost ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
172.17.0.2   site1_ucalgary_win11
192.168.2.100 server
```

4. Launch `site1_ucalgary_win11` by running:

```
chmod -R 777 /workspace/my-workspace
/workspace/my-workspace/site1_ucalgary_win11/startup/start.sh
```

```
root@00bf59daa070:/workspace/my-workspace# chmod -R 777 /workspace/my-workspace
root@00bf59daa070:/workspace/my-workspace# /workspace/my-workspace/site1_ucalgary_win11/startup/start.sh
root@00bf59daa070:/workspace/my-workspace# WORKSPACE set to /workspace/my-workspace/site1_ucalgary_win11/startup/
..
PYTHONPATH is /local/custom:
start fl because of no pid.fl
```

5. `server` - `site1_ucalgary_win11` connection can be verified through info from terminal.

```
2024-07-10 07:29:05,996 - FederatedClient - INFO - Successfully registered client:site1_ucalgary_win11 for projec
t Diabetes Demo. Token:31857ac6-f9fc-4012-a29c-f8e35fc0cdee SSID:ebc6125d-0a56-4688-9b08-355fe9e4d61a
2024-07-10 07:38:19,602 - ClientEngine - INFO - Starting client app. rank: 0
```

6. Open a new terminal, switch to root user by running `sudo su`. Launch `site2_ucalgary_win11` by repeating previous step 1-5 with site name changed to `site2_ucalgary_win11`.

```
sudo su
docker exec -it site2_ucalgary_win11 /bin/bash
ls
nano /etc/hosts
```

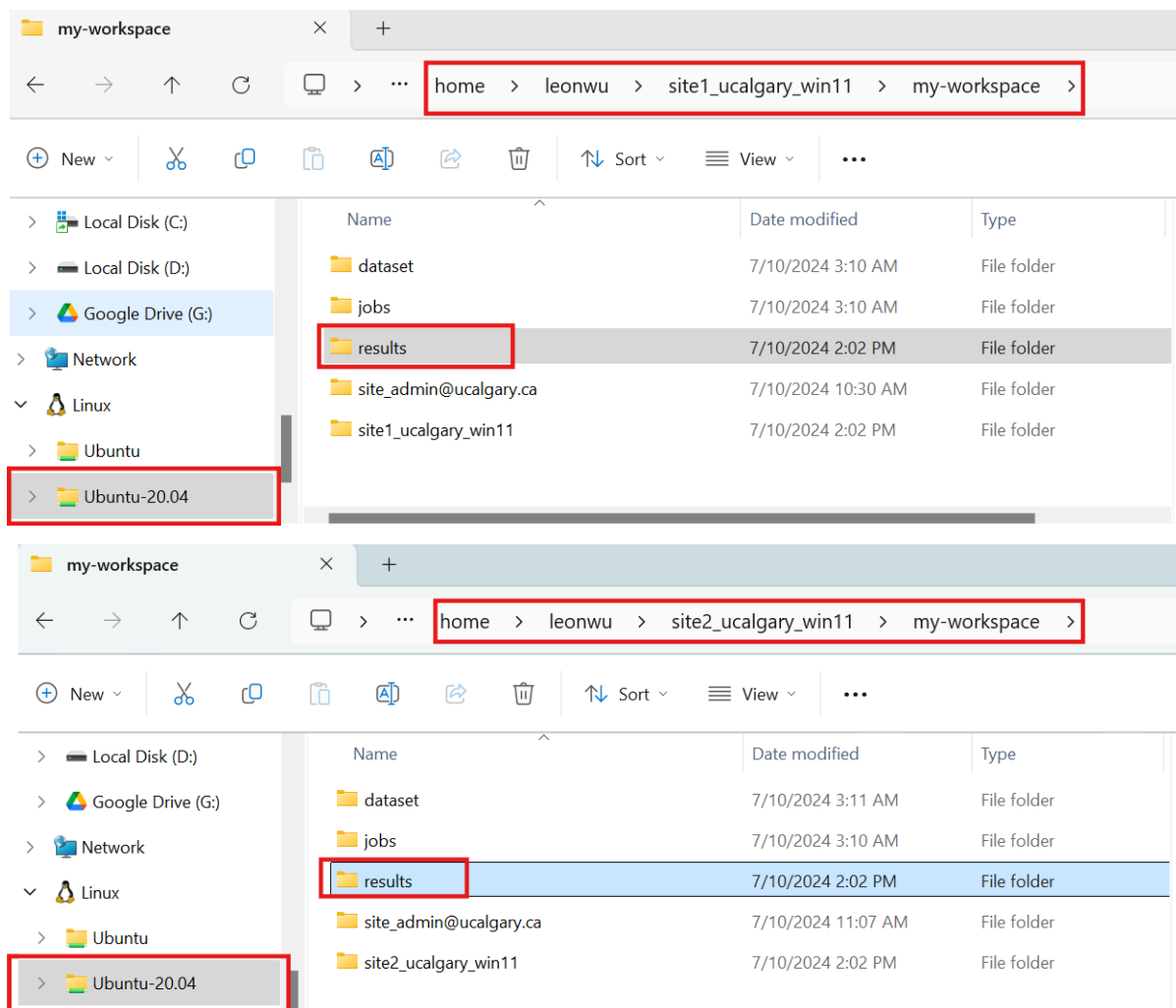
```
GNU nano 4.8 /etc/hosts
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.3 site2_ucalgary_win11
192.168.2.100 server
```

```
cat /etc/hosts
chmod -R 777 /workspace/my-workspace
/workspace/my-workspace/site2_ucalgary_win11/startup/start.sh
```

```
Leonwu@DESKTOP-AI:~$ sudo su
root@DESKTOP-AI:/home/leonwu# docker exec -it site2_ucalgary_win11 /bin/bash
root@5f3162866e70:/workspace/my-workspace# ls
Diabetes Demo site2_ucalgary_win11 site2_ucalgary_win11/startup
root@5f3162866e70:/workspace/my-workspace# nano /etc/hosts
root@5f3162866e70:/workspace/my-workspace# cat /etc/hosts
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.3 site2_ucalgary_win11
192.168.2.100 server
root@5f3162866e70:/workspace/my-workspace# chmod -R 777 /workspace/my-workspace
root@5f3162866e70:/workspace/my-workspace# /workspace/my-workspace/site2_ucalgary_win11/startup/start.sh
root@5f3162866e70:/workspace/my-workspace# WORKSPACE set to /workspace/my-workspace/site2_ucalgary_win11/startup/..
PYTHONPATH is /local/custom:
start fl because of no pid.fl
new pid 616
Waiting for SP...
2024-07-10 17:54:08,370 - CoreCell - INFO - site2_ucalgary_win11: created backbone external connector to grpc
://server:8002
2024-07-10 17:54:08,370 - ConnectorManager - INFO - 616: Try start_listener Listener resources: {'secure': False, 'host': 'localhost'}
2024-07-10 17:54:08,371 - nvflare.fuel.f3.sfm.conn_manager - INFO - Connector [CH00002 PASSIVE tcp://0:34208] is starting
2024-07-10 17:54:08,872 - CoreCell - INFO - site2_ucalgary_win11: created backbone internal listener for tcp://localhost:34208
2024-07-10 17:54:08,872 - nvflare.fuel.f3.sfm.conn_manager - INFO - Connector [CH00001 ACTIVE grpc://server:8002] is starting
2024-07-10 17:54:08,874 - FederatedClient - INFO - Wait for engine to be created.
2024-07-10 17:54:08,881 - nvflare.fuel.f3.drivers.grpc_driver.GrpcDriver - INFO - created secure channel at server:8002
2024-07-10 17:54:08,881 - nvflare.fuel.f3.sfm.conn_manager - INFO - Connection [CN00002 N/A => server:8002] is created: PID: 616
2024-07-10 17:54:09,298 - FederatedClient - INFO - Successfully registered client:site2_ucalgary_win11 for project Diabetes Demo. Token:72ca5a2b-64b9-4811-baea-e6d1e4d3b676 SSID:ebc6125d-0a56-4688-9b08-355fe9e4d61a
```

Access Results

Once a job is completed, results are saved at `results` folder in each site's `my-workspace` folder. Results include local and global reports on each round, as well as a `results.csv` containing local and global AUC values on each round.



For macOS

Install Docker Desktop

1. Download [Docker Desktop for Mac](#).

This page contains download URLs, information about system requirements, and instructions on how to install Docker Desktop for Mac.

Docker Desktop for Mac with Apple silicon

Docker Desktop for Mac with Intel chip

2. Install Docker Desktop:

- Once the download is complete, open the downloaded `.dmg` file.
- In the Finder window that appears, drag the Docker icon to the Applications folder. This will install Docker Desktop.

3. Launch Docker Desktop:

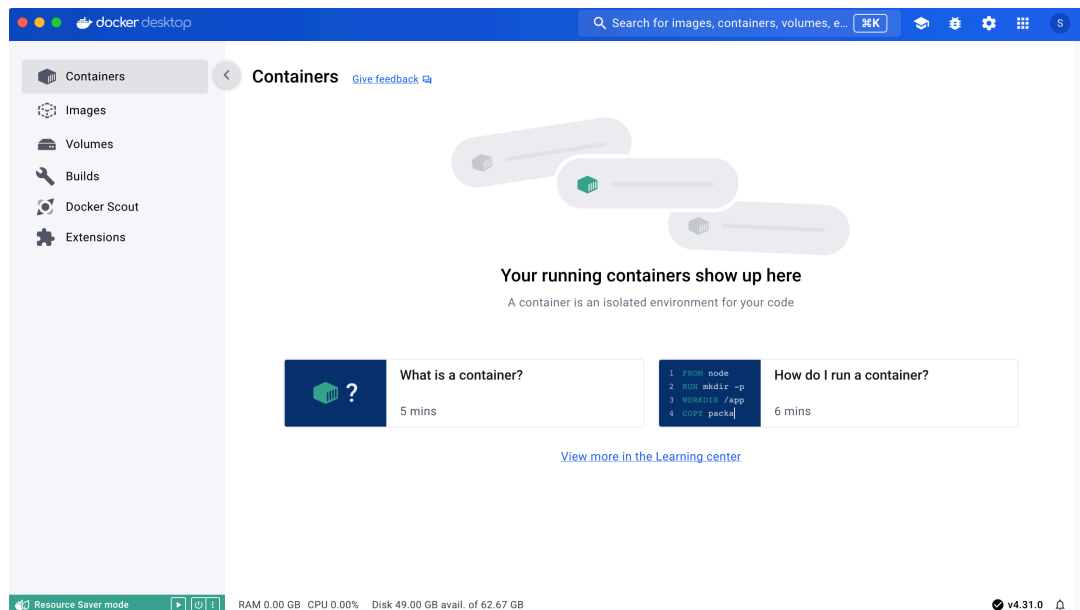
- Open the Applications folder and double-click the Docker icon to launch Docker Desktop.



- The Docker whale icon will appear in the menu bar and a dialog will pop up asking you to authorize Docker Desktop with your system password. Enter your password and click **OK**.

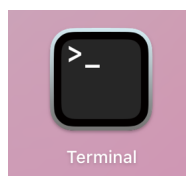
4. Complete Initial Setup:

- Docker Desktop will start and you will see a welcome message.
- Follow any additional prompts to complete the initial setup. Docker Desktop will start the Docker daemon automatically.



5. Verify Installation:

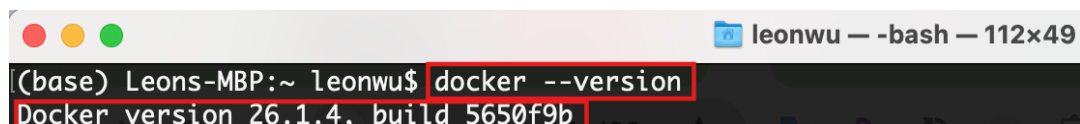
- Open a Terminal window from the Applications folder under Utilities.



- Run the following command to check that Docker is installed and running correctly:

```
docker --version
```

- You should see output similar to `Docker version XX.XX.XX, build xxxxxxxx`.



Download Nvflare_mac Image From Dockerhub

1. In terminal, download [nvflare_mac](#) docker image by running:

```
docker pull shuiliangwu/nvflare_mac:latest
```

2. Verify the download of nvflare_mac image by running:

```
docker images
```

```
(base) Leons-MBP:~ leonwu$ docker pull shuiliangwu/nvflare_mac:latest
latest: Pulling from shuiliangwu/nvflare_mac
9ea8908f4765: Pull complete
75ffc57cea9c: Pull complete
5c8fc1c86705: Pull complete
2f83981de011: Pull complete
Digest: sha256:52f891f94ffee46ea1c0c2ce9e746efa98b512ccfcf3c7951dcb0ed9fa395211
Status: Downloaded newer image for shuiliangwu/nvflare_mac:latest
docker.io/shuiliangwu/nvflare_mac:latest
(base) Leons-MBP:~ leonwu$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
shuiliangwu/nvflare_mac  latest     4e904631e435  3 days ago   9.25GB
```

Setup Workspaces

1. For the sake of convenience, let's create Site docker containers in Desktop folder by running
(For any scripts in this Guide that contain the placeholder `leonwu`, make sure to replace `leonwu` with your own Mac account name before using them):

```
mkdir /Users/leonwu/Desktop/site_stjoes_mac
mkdir /Users/leonwu/Desktop/site_stjoes_mac/my-workspace
chmod -R 777 /Users/leonwu/Desktop/site_stjoes_mac/my-workspace
```

```
(base) Leons-MBP:~ leonwu$ mkdir /Users/leonwu/Desktop/site_stjoes_mac
(base) Leons-MBP:~ leonwu$ mkdir /Users/leonwu/Desktop/site_stjoes_mac/my-workspace
(base) Leons-MBP:~ leonwu$ chmod -R 777 /Users/leonwu/Desktop/site_stjoes_mac/my-workspace
```

We shall see `site_stjoes_mac` folder in Desktop with `my-workspace` subfolder in it.

2. Copy below four folders into new created `my-workspace` folder.
- `site_admin@stjoes.ca` folder: site FLARE console kit (from project manager)
 - `site_stjoes_mac` folder: site startup kit (from project manager)
 - `jobs` folder: contains source code and configuration files (from project manager)
 - `dataset` folder: contains your dataset

my-workspace								
Name		Date Modified	Size					
dataset		Yesterday at 9:33 PM						
output		Yesterday at 9:34 PM						
site_stjoes_mac_header.csv		Jul 5, 2024 at 2:34 AM	1					
site_stjoes_mac.csv		Jul 5, 2024 at 2:34 AM						
jobs		Today at 1:46 AM						
logistic_learner		Today at 1:46 AM						
app		Today at 1:46 AM						
config		Yesterday at 1:58 AM						
config_fed_client.conf		Jul 7, 2024 at 10:45 PM						
config_fed_server.conf		Yesterday at 1:58 AM						
custom		Yesterday at 1:51 AM						
logistic_learner.py		Jun 21, 2024 at 10:08 AM						
model_config.json		Yesterday at 1:51 AM	32					
meta.conf		Jul 7, 2024 at 10:45 PM	10					
site_admin@stjoes.ca		Jul 7, 2024 at 1:42 AM						
site_stjoes_mac		Yesterday at 9:52 PM						

Macintosh HD > Users > leonwu > Desktop > site_stjoes_mac > my-workspace

Create Site Docker Container(s)

1. Go to `site_stjoes_mac` folder by running:

```
cd /Users/leonwu/Desktop/site_stjoes_mac
```

We shall see our current location is `site_stjoes_mac`.

It is crucial that we are at above location in order to mount workspace.

```
(base) Leons-MBP:~ leonwu$ cd /Users/leonwu/Desktop/site_stjoes_mac
(base) Leons-MBP:site_stjoes_mac leonwu$
```

2. Create Site docker container by running:

```
docker run --privileged --rm -d -it --ipc=host \
--ulimit memlock=-1 --ulimit stack=67108864 \
-v "$(pwd -P)/my-workspace:/workspace/my-workspace" \
-v /var/run/docker.sock:/var/run/docker.sock \
-w /workspace/my-workspace \
--name site_stjoes_mac \
shuiliangwu/nvflare_mac:latest
```

Above site name `site_stjoes_mac` shall be replaced with the actual ones from your setup.

```
(base) Leons-MBP:site_stjoes_mac leonwu$ docker run --privileged --rm -d -it --ipc=host \
> --ulimit memlock=-1 --ulimit stack=67108864 \
> -v "$(pwd -P)/my-workspace:/workspace/my-workspace" \
> -v /var/run/docker.sock:/var/run/docker.sock \
> -w /workspace/my-workspace \
[> --name site_stjoes_mac \
> shuiliangwu/nvflare_mac:latest
92b43c243d2c08177b4968a7e998f33d33bbe59ee077231ba896639b4e5ffc86
```

3. Verify docker containers by running:

```
docker ps -a
```

```
(base) Leons-MBP:site_stjoes_mac leonwu$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS   NAMES
92b43c243d2c   shuiliangwu/nvflare_mac:latest     "/bin/bash"             17 seconds ago Up 16 seconds           site_stjoes_mac
```

Launch NVFLARE Site(s)

1. Access to Site docker container by running:

```
docker exec -it site_stjoes_mac /bin/bash
```

```
(base) Leons-MBP:site_stjoes_mac leonwu$ docker exec -it site_stjoes_mac /bin/bash
root@92b43c243d2c:/workspace/my-workspace# ls
```

2. We shall see previous copied four folders by running:

```
ls
```

```
[root@92b43c243d2c:/workspace/my-workspace# ls
dataset  jobs  site_admin@stjoes.ca  site_stjoes_mac
```

3. Modify the `/etc/hosts` file of Site docker container.

- Open `/etc/hosts` file by running:

```
nano /etc/hosts
```

- Replace the hostname of `172.17.0.2` by `site_stjoes_mac`, and add `server`'s ip address and hostname.

```
127.0.0.1    localhost
::1         localhost ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
172.17.0.2   site_stjoes_mac
192.168.2.100 server
```

- Press `Ctrl + O`, then `Enter` to save the change. Press `Ctrl + X` to exit.
- Verify the modification by running:

```
cat /etc/hosts
```

```

root@92b43c243d2c:/workspace/my-workspace# cat /etc/hosts
127.0.0.1    localhost
::1         localhost ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
172.17.0.2   site_stjoes_mac
192.168.2.100 server

```

4. Launch `site_stjoes_mac` by running:

```
/workspace/my-workspace/site_stjoes_mac/startup/start.sh
```

5. `server` - `site_stjoes_mac` connection can be verified through info from terminal.

```

root@92b43c243d2c:/workspace/my-workspace/site_stjoes_mac/startup# WORKSPACE set to /workspace/my-workspace/site_stjoes_mac/startup/..
PYTHONPATH is /local/custom:
start fl because of no pid.fl
new pid 36
Waiting for SP...
2024-07-09 01:39:20,152 - CoreCell - INFO - site_stjoes_mac: created backbone external connector to grpc://server:8002
2024-07-09 01:39:20,153 - ConnectorManager - INFO - 36: Try start_listener Listener resources: {'secure': False, 'host': 'localhost'}
2024-07-09 01:39:20,155 - nvflare.fuel.f3.sfm.conn_manager - INFO - Connector [CH00002 PASSIVE tcp://0:21336] is starting
2024-07-09 01:39:20,662 - CoreCell - INFO - site_stjoes_mac: created backbone internal listener for tcp://localhost:21336
2024-07-09 01:39:20,663 - nvflare.fuel.f3.sfm.conn_manager - INFO - Connector [CH00001 ACTIVE grpc://server:8002] is starting
2024-07-09 01:39:20,674 - FederatedClient - INFO - Wait for engine to be created.
2024-07-09 01:39:20,740 - nvflare.fuel.f3.drivers.grpc_driver.GrpcDriver - INFO - created secure channel at server:8002
2024-07-09 01:39:20,741 - nvflare.fuel.f3.sfm.conn_manager - INFO - Connection [CN00002 N/A => server:8002] is created: PID: 36
2024-07-09 01:39:20,988 - FederatedClient - INFO - Successfully registered client:site_stjoes_mac for project Diabetes Demo. Token:0f4306be-699e-4101-b3c6-370b860a66d5
2024-07-09 01:39:20,988 - FederatedClient - INFO - Got engine after 0.32324767112731934 seconds
2024-07-09 01:39:20,999 - FederatedClient - INFO - Got the new primary SP: grpc://server:8002

```

Access Results

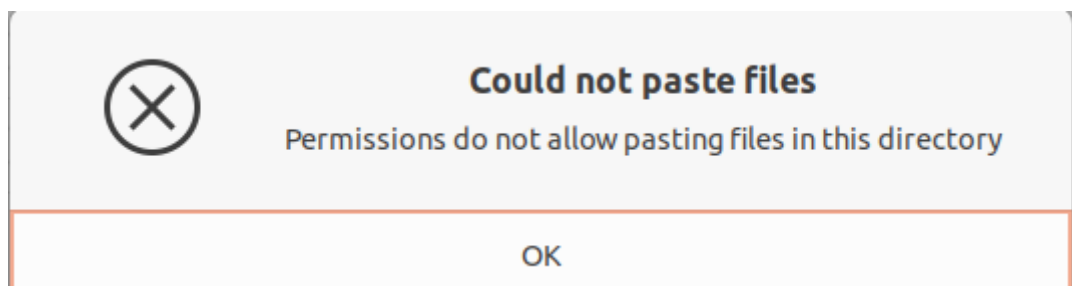
Once a job is completed, results are saved at `results` folder in each site's `my-workspace` folder. Results include local and global reports on each round, as well as a `results.csv` containing local and global AUC values on each round.

my-workspace				
Name				Date Modified
> dataset				Yesterday at 9:33 PM
> jobs				Today at 1:46 AM
▼ results				Yesterday at 11:03 PM
▼ logistic_regression_07092024_015221				Yesterday at 9:52 PM
global_report_0_site_stjoes_mac.txt				Yesterday at 9:52 PM
global_report_1_site_stjoes_mac.txt				Yesterday at 9:52 PM
global_report_3_site_stjoes_mac.txt				Yesterday at 9:52 PM
global_report_4_site_stjoes_mac.txt				Yesterday at 9:52 PM
global_report_5_site_stjoes_mac.txt				Yesterday at 9:52 PM
global_report_6_site_stjoes_mac.txt				Yesterday at 9:52 PM
global_report_7_site_stjoes_mac.txt				Yesterday at 9:52 PM
global_report_8_site_stjoes_mac.txt				Yesterday at 9:52 PM
global_report_9_site_stjoes_mac.txt				Yesterday at 9:52 PM
local_report_0_site_stjoes_mac.txt				Yesterday at 9:52 PM
local_report_1_site_stjoes_mac.txt				Yesterday at 9:52 PM
local_report_3_site_stjoes_mac.txt				Yesterday at 9:52 PM
local_report_4_site_stjoes_mac.txt				Yesterday at 9:52 PM
local_report_5_site_stjoes_mac.txt				Yesterday at 9:52 PM
local_report_6_site_stjoes_mac.txt				Yesterday at 9:52 PM
local_report_7_site_stjoes_mac.txt				Yesterday at 9:52 PM
local_report_8_site_stjoes_mac.txt				Yesterday at 9:52 PM
local_report_9_site_stjoes_mac.txt				Yesterday at 9:52 PM
results.csv				Yesterday at 9:52 PM
> site_admin@stjoes.ca				Jul 7, 2024 at 1:42 AM
> site_stjoes_mac				Yesterday at 9:52 PM

Troubleshooting

Common Issues

- **"Permissions do not allow ..."**: This kind of pop-up window during your operations in Ubuntu indicates you need to change the permissions of files or directories in Ubuntu.



Solution: `chmod 777` command.

- Purpose: Sets read, write, and execute permissions for everyone.
- Usage:
 - For files such as `myfile.txt`: `chmod 777 myfile.txt`

- For directories such as `/home/leon/workspace/`: `chmod -R 777 /home/leon/workspace/`
 - Effect: Grants read (`r`), write (`w`), and execute (`x`) permissions to the user (owner), group, and others.
- `permission denied while trying to connect to the Docker daemon socket...`: Switched to a root user by running `sudo su` and input your account password.

```
leonwu@DESKTOP-AI:~$ docker images
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/images/json": dial unix /var/run/docker.sock: connect: permission denied
leonwu@DESKTOP-AI:~$ sudo su
[sudo] password for leonwu:
root@DESKTOP-AI:/home/leonwu# docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
shuiliangwu/nvflare_mac  latest     4e904631e435  4 days ago   9.25GB
```

- `There seems to be one instance, pid=###, running.`: This error happens when you are trying to launch Site(s) while it is running at background.
 - Solution 1: solve it by following instructions from the error by running:

```
# replace ### with actual pid
kill -9 ###

# replace your-actual-site-name with corrected name
rm your-actual-site-name/startup/daemon_pid.fl
```

```
root@0e572f870d24:/workspace/my-workspace# ./server/startup/start.sh
root@0e572f870d24:/workspace/my-workspace# WORKSPACE set to /workspace/my-workspace
./server/startup/..
There seems to be one instance, pid=7567, running.
If you are sure it's not the case, please kill process 7567 and then remove daemon_pid.fl in /workspace/my-workspace/server/startup/..
kill -9 7567
root@0e572f870d24:/workspace/my-workspace# rm server/startup/daemon_pid.fl
```

- Solution 2: If solution 1 doesn't work, **send a request to project manager to restart the server/client in FLARE console.**

Support Resources

- For more information on WSL2 Installation, visit [WSL Installation Manual](#).
- For more information on Docker, visit [Docker Manuals](#).
- For more information on CUDA, visit [CUDA on WSL User Guide](#).
- For more information on NVIDIA Container Toolkit, visit [NVIDIA Container Toolkit Installation Guide](#).
- For more information on Docker Utility Engine for NVIDIA GPUs, visit [Docker Utility Engine for NVIDIA GPUs](#).
- For more information on NVFLARE, visit [NVFLARE Official Documents](#).

Appendix

Requirements of Data Format

After raw data cleaning, cleaned data (`.csv` or `.xlsx`) shall follow requirements below in order to be recognized by the source code for federated learning.

1. **Relocate Target Feature.** The target feature and its data (also known as the dependent attribute/variable or response attribute/variable in data science) shall be relocated to the first column of the datasheet. All the sites shall have the same target feature.
2. **Data Encoding.** Each column in the raw data must be encoded into a numerical format suitable for machine learning. All the sites shall follow the same **Data Encoding Rule (Data Encoding Rule** for this demo can be seen [here](#)).
 - **Categorical Encoding:** Categorical variables should be converted to numerical values using appropriate encoding techniques like one-hot encoding or label encoding.
 - **Numerical Columns:** Continuous numerical values should remain numerical but may require normalization or scaling.
3. **File Format.** The final dataset folder shall contains two files in CSV format.
 - `{site_name}.csv`: this number-only file contains all encoded data without a header row.
 - `{site_name}_header.csv`: this one-row file contains all feature names (header row) with column-by-column matching with data in `{site_name}.csv`. This file shall be identical for all sites.

`{site_name}` shall be replaced by the actual site name. For example, `site_server_win11.csv` and `site_server_win11_header.csv` in this demo.

Example of cleaned data:

	A	B	C	D	E	F	G	H	I
1	race	gender	age	time_in_hospital.x	HbA1c	diabetesMed	readmitted	admission_source_id	number_outpatient_emerg_inpatient
2	Caucasian	Female	[80-90)	4	>7	Yes	NO	Emergency room	0
3	Caucasian	Female	[80-90)	3	>8	Yes	NO	Emergency room	0
4	Caucasian	Female	[70-80)	12	Norm	Yes	NO	Physician/clinic referral	0
5	Caucasian	Male	[60-70)	8	>8	Yes	NO	Physician/clinic referral	0
6	Caucasian	Male	[70-80)	11	>8	Yes	>30	Emergency room	2
7	Caucasian	Male	[70-80)	7	>8	Yes	>30	Physician/clinic referral	0
8	AfricanAmerican	Male	[50-60)	2	Norm	Yes	NO	Physician/clinic referral	0
9	AfricanAmerican	Female	[80-90)	7	>8	Yes	>30	Emergency room	0
10	Caucasian	Male	[50-60)	13	Norm	Yes	NO	Emergency room	0
11	Caucasian	Female	[70-80)	6	Norm	Yes	>30	Emergency room	1
12	Caucasian	Female	[70-80)	6	>8	No	>30	Emergency room	0
13	Caucasian	Female	[70-80)	9	Norm	Yes	NO	Emergency room	1
14	Caucasian	Male	[80-90)	7	>7	Yes	NO	Other	0
15	Caucasian	Male	[50-60)	5	>7	Yes	NO	Physician/clinic referral	0
16	AfricanAmerican	Female	[50-60)	2	>8	Yes	>30	Other	0
17	AfricanAmerican	Female	[40-50)	4	Norm	Yes	NO	Emergency room	0
18	Caucasian	Male	[70-80)	8	>8	Yes	>30	Physician/clinic referral	0
19	Caucasian	Male	[70-80)	12	>8	Yes	>30	Physician/clinic referral	0
20	Caucasian	Female	[10-20)	3	>8	Yes	>30	Emergency room	0

Example of `{site_name}_header.csv` with `readmitted` as target feature:

	A	B	C	D	E	F	G	H	I
1	readmitted	race	gender	age	time_in_hospital.x	HbA1c	diabetesMed	admission_source_id	number_outpatient_emerg_inpatient
2									
3									
4									
5									

Example of `{site_name}.csv` with `readmitted` as target feature:

	A	B	C	D	E	F	G	H	I
1	0	3	0	8	4	1	1	0	0
2	0	3	0	8	3	2	1	0	0
3	0	3	0	7	12	0	1	2	0
4	0	3	1	6	8	2	1	2	0
5	1	3	1	7	11	2	1	0	2
6	1	3	1	7	7	2	1	2	0
7	0	1	1	5	2	0	1	2	0
8	1	1	0	8	7	2	1	0	0
9	0	3	1	5	13	0	1	0	0
10	1	3	0	7	6	0	1	0	1
11	1	3	0	7	6	2	0	0	0
12	0	3	0	7	9	0	1	0	1
13	0	3	1	8	7	1	1	1	0
14	0	3	1	5	5	1	1	2	0
15	1	1	0	5	2	2	1	1	0
16	0	1	0	4	4	0	1	0	0
17	1	3	1	7	8	2	1	2	0
18	1	3	1	7	12	2	1	2	0
19	1	3	0	1	3	2	1	0	0
20	1	3	1	5	4	2	1	0	0

A demo of data formatting can be seen [here](#) in the folder of `data_formatting_demo`.